Bayesian Real-time Optical Flow

John S. Zelek
School of Engineering, Univ. of Guelph
Guelph, ON, N1G 2W1, Canada
jzelek@uoguelph.ca

Abstract

Optical flow can be used to compute motion detection, time to collision, structure, focus of expansion as well as object segmentation. Unfortunately, most optical flow techniques do not provide accurate and dense measures that are useful for these types of computations. In addition, most techniques are also slow computationally. Albeit, one method proposed by Camus is able to perform optical flow computations in real-time capitalizing on redundancies in the computation and spatial-temporal sampling trade-offs. It is a simple technique based on simulating various motions and computing the SD (sum-difference) of patches. Its problem is that the produced field is not accurate and arbitrary in aperture and blank wall situations. We show that the simulating of various futures can be used as the factored samples that produce the likelihood probabilities that can be used in a particle filtering framework. Maximization/minimization or computing the expectations of the likelihood at a particular location does not necessarily produce the proper flow. We suggest that likelihoods are well behaved when their variance is small and these can be propagated firstly to address aperture problems and secondly to address the extended blank wall problem. We show this propagation with thresholded likelihood values and speculate on how the likelihood distributions can be integrated into an algorithm that has its basis in particle filtering.

1 Introduction

Optical flow is what results from the recovery of the 2-D motion field (i.e., the projection of the 3D velocity profile onto a 2-D plane; or the resulting apparent motion in an image). Most optical flow techniques assume that uniform illumination is present and that all surfaces are Lambertian. Obviously this does not necessarily hold in the real-world, but we assume that these conditions do hold locally. Optical flow describes the direction and speed of feature motion in the 2D image as a result of relative motion between the viewer and the scene. If the camera is fixed, the motion can be attributed to the moving objects in the scene. Optical flow also encodes useful information about scene structure: e.g., distant objects have much slower apparent motion than close objects. The apparent motion of objects on the image plane provides strong cues for interpreting structure and 3-D motion. Some creatures in nature such as birds are chiefly reliant on motion cues for understanding the world.

Optical flow may be used to compute motion detection, time-to-collision, focus of expansion as well as object segmentation; however, most optical flow techniques do not produce an accurate flow map necessary for these calculations [1]. Most motion techniques make the assumption that image irrandiance remains constant during the motion process. The optical flow equation relates temporal (I_t) changes in image intensity ($I(x, y, t)$) to the velocity (i.e., disparity) ((u, v)).

$$I_x u + I_y v + I_t = 0 \quad (1)$$

This equation is not well posed and many approaches [2] use a smoothness constraint to render the problem well-posed.

$$E^2(x, y) = (I_x u + I_y v + I_t)^2 + \lambda({u_x}^2 + {u_y}^2 + {v_x}^2 + {v_y}^2) \quad (2)$$

Motion field computations are similar to stereo disparity measures albeit for the spatial differences being smaller between temporal images (because of a high sampling rate) and the 3-D displacement between the camera and the scene not necessarily being caused by a single 3D rigid transformation.

Motion recovery techniques have been classified [3] into *intensity-based differential methods*, *frequency-based filtering methods* and *correlation-based methods*. In addition, most of the approaches have usually three steps of processing [3]: (1) *prefiltering* (low or band pass) to enhance signal-to-noise; (2) *measurement extraction* such as spatiotemporal derivatives or local correlation surfaces; and (3) *measurement integration* by regularization, correlation or a least-squares computation. Derivatives can be difficult to compute, especially temporal derivatives where values into the future are not available. Other problems that have been discussed in survey papers [3] include: (1) the validity of equating optical flow with image motion; (2) the problem

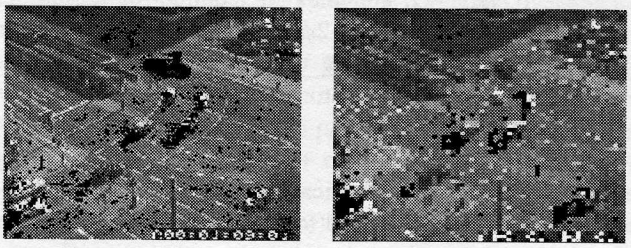
posed by occluding surfaces; and (3) the problem of transparent motions. Optical flow only approximates image flow and is different when texture is absent (i.e., optical flow is zero) and when the true motion field violates the brightness consistency model (e.g., highlights, shadows, variable illumination). Studies of evaluating optical flow methods [1], [4] have showed that differential and phase-based (frequency category) methods produced better results than correlation based methods, but still the results are problematic especially for the extraction of additional information (e.g., structure, segmentation, etc.).

One method of computing optical flow that interested us was a simple method of exhaustive search over a small neighbourhood across n previous frames [5]. We found this technique performed well when used to segment moving from stationary objects however typically the optical field values were flawed. Figure 1 shows the results of segmenting moving from static objects based on temporally accumulating optical flow using a simple SD (sum-difference) computation [5]. A potential function is created by summing directional optical flow vectors temporally. Oscillating objects (e.g., tree branches) would evolve into small potentials over time while objects that traverse the scene (e.g., cars) would produce a high potential value.

The SD technique [5] is intriguing because redundancies in the calculations can be minimized to to speed performance. We empirically investigated performance to be in the order of video frame capture rates (i.e., 10 to 30 fps) for 160x120 images on a Pentium II machine. The authors claim that accuracy-performance trade-off is such that this technique achieves reasonable accuracy at substantial better speeds when compared to other techniques, especially when compared to differential and frequency-based methods. Typically, we found the performance claim validated but we encountered poor accuracy. The contribution of this technique was the spatial-temporal search trade-off, where search temporally is $O(n)$ as opposed to $O(n^2)$ spatially. The exhaustive element of the algorithm is also interesting in that many possible futures are predicted. This led us to explore a Bayesian interpretation of the search as providing posterior probabilities for the optical flow field.

2 Real-time Optical Flow

The algorithm we used for computing optical flow information is a correlation-based technique (based on simulating various motions) that has been shown to exhibit real-time performance [5]. In this technique, the motion of a pixel at $[x, y]$ in one frame to a successive frame, is defined by the determined motion of a patch P_μ of μ by μ pixels centred at $[x, y]$, out of $(2n + 1) * (2n + 1)$ possible displacements, where n is an arbitrary parameter dependent on the maxi-



(a) (b)

Figure 1: **Traffic Scene:** (a) and (b) show one temporal shot of a sequence of traffic frames at different resolutions. The motion patterns, as indicated by black pixels, are readily visible.

mum expected motion in the image over two successive image frames in a temporal sequence. The motion of the patch is simulated for each potential displacement of $[x, y]$ (given by n) and a match strength M is calculated for each displacement. Let ϕ represent a matching function. If I_1 is the first image examined and I_2 is the next successive image in a temporal sequence, then the match strength for a patch $[x, y]$ for a simulated displacement (u, v) is calculated by:

$$\begin{aligned} &\forall u, w : M(x, y; u, v) = \\ &\textstyle\sum \phi(I_1(i, j) - I_2(i + u, j + v)), (i, j) \in P_\mu \end{aligned} \tag{3}$$

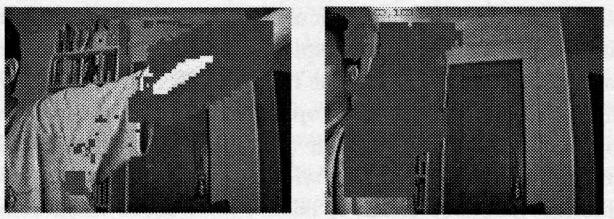
This can be efficiently calculated by taking into account certain redundancies in the calculation. Let m represent the simulation of all possible displacements of pixels $[x, y]$ between I_1 and I_2, and M be the function that results by applying ϕ (which is a smoothing operator) over m. The smoothing is only done over similar (u, v) displacements. In addition, since most averaging windows share common values amongst neighbours, this is taken into account when computing m and subsequently M to reduce the computational time.

Additional efficiency in the algorithm can be obtained by controlling the spatial or temporal sampling [5]. Spatial sampling is constrained by the size of the figure of interest and temporal sampling is constrained by the top speed of the figure of interest. It is also claimed [5] that sampling at various resolutions and interpolating the results can also result in minimizing the effects of occlusion, however we were not able to verify this claim. The computational efficiency of the approach - $O(s)$, s is the number of pixels - is at the expense of an increase in storage capacity. In particular (where s is a single dimension of a square image, $(2n + 1)^2$ elements define the square plausible pixel displacement region, and $(2\phi_n + 1)^2$ define the square smoothing window), s^2 elements are required for storing each temporal image (minimally two images), and for each pair of temporally displaced images the following is required:

$(s-2\phi_n)^2(2n+1)^2$ elements are required for storing the simulated motion array; $(s-2\phi_n)^2$ elements are required for storing temporary smoothing values; and $2(s-2\phi_n)^2$ elements are also required for storing the resulting optical flow vector field. With regards to computation, the basic algorithm requires $(s-2\phi_n)[(2n+1)^2(s+1)+4s-6\phi_n-3]$ additions/subtractions. Typically both ϕ_n and n are relatively small when compared to s resulting in an algorithm complexity of $O(s^2)$.

We found that the Camus algorithm is problematic in that it does not typically produce an accurate dense flow field. This is especially true for objects where the intensity slowly changes across the object. In order to partially rectify this and still obtain reasonable performance, we solved Laplace's equation in each horizontal row with Dirichlet boundary conditions imposed on the bounds as well as the so-called reliable computed optical flow vectors. The rationale was to mimic at a reduced computational cost, the smoothing criteria that the energy based methods use. It should be noted that we estimated a noise value in our image and used it as a threshold to remove noisy flow vectors. Unfortunately, the threshold values were scene dependent.

Figure 2 shows the results of using only two adjacent temporal scenes for segmenting moving from static regions. Optical flow vectors are classified as reliable or not and the reliable vectors are diffused (discussed later).



(a) (b)

Figure 2: **Dense Flow Estimate:** (a) shows where optical flow vectors were detected using the Camus algorithm [5], while (b) shows the results of the same algorithm with post processing to produce a dense flow.

3 Velocity Likelihoods

A recent hypothesis [6] is that early motion analysis is the extraction of local likelihoods which are subsequently combined with the observer's prior assumptions to estimate object motion. Ambiguity is present in the local motion information, either as a result of the *aperture problem* (e.g., the vertical motion component is not attainable from a horizontally moving edge just based on local information) [7] or the *extended blank wall problem* (i.e., both vertical and horizontal gradients are zero and many motion velocities (u,v) fit the brightness constancy equation) [8].

The goal in a Bayesian approach to motion analysis is to calculate the posterior probability of a velocity given the image data [6]. The posterior probability is computing using the spatiotemporal brightness observation (i.e., measurement) $I(x,y,t)$ at location x,y and time t and the 2D motion (u,v) of the object, where α is a normalization constant independent of (u,v):

$$P(u,v \mid I(x,y,t)) = \alpha P(u,v) P(I(x,y,t) \mid u,v) \quad (4)$$

Assuming that the image observations at different positions and times are conditionally independent, given u,v, then:

$$P(I(x,y,t) \mid u,v) = \alpha P(u,v) \prod_{i,j} P(I(x_i,y_i,t_j) \mid u,v) \quad (5)$$

where the product is taken over all positions x_i, y_i and times t_j.

The quantity to compute is the likelihood of a velocity $P(I(x_i,y_i,t_j) \mid u,v)$. This also assumes that we are only concerned with a single object which many not necessarily be the case. $P(u,v)$, the prior, has been hypothesized [6] that it should favour slow speeds.

There have been three distinct approaches for image velocity likelihood functions. The first two look at various contaminations of the optical flow constraint equation.

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (6)$$

The first approach assumes that the temporal derivative is contaminated with mean-zero Gaussian noise, but the spatial derivative measurements are noise-free [9]. This results in a likelihood that resembles a *fuzzy constraint line* and is defined as:

$$P(I \mid u,v) = \alpha \exp \frac{-1}{2\sigma^2} \int (\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t})^2 dxdt \quad (7)$$

The second approach assumes that mean-zero Gaussian noise is added to all of the spatial and temporal derivative measurements [10]. This results in a likelihood that looks like a *fuzzy bowtie*.

$$P(I \mid u,v) = \alpha \exp \frac{-1}{2\sigma^2} \frac{\int (\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t})^2}{1+v^2+u^2} dxdt \quad (8)$$

There appears to be no consensus regarding what likelihood to use [6], and others have even suggested a likelihood where the noise in the spatial and temporal derivatives is correlated $E(I_x I_y I_t) = -\sigma_{xyt} sgn(I_x I_y I_t)$ [11]. Alternatively, generative models have also been proposed [6] and models

[12] (here, state is expressed as translation and occlusion situations as opposed to just velocities) which use the Condensation algorithm [13]. The generative model [6] proposed, uses a model of a scene and noise and calculates the predicted intensity assuming a scene function moving at that velocity. The residual intensity is attributed to noise, and the less energy in the residual, the more likely the velocity. The energy is converted to a probability that is a weighted exponential of the residual. Extensions are presented where SSD (sum-squared difference) and gradient constraint methods are shown to perform similar computations. In the case of SSD, minimizing the SSD is equivalent to maximizing the likelihood. Also, minimizing the gradient constraint is also comparable to maximizing the likelihood.

Equivalently, we argue that SD (sum difference) can also be expressed as a likelihood. Thus making the simplistic approach proposed by Camus [5] a candidate algorithm for a Bayesian approach for real-time optical flow computation. Rather than computing a single likelihood for the scene, we compute a likelihood for each overlapping patch. We also argue that there are really three different cases: (1) a well defined symmetric likelihood; (2) an anti-symmetrical likelihood (i.e., aperture problem), and (3) a flat likelihood (i.e., extended blank wall or zero flow). We postulate that the shape of the likelihood (i.e., variance) is an indicator of the reliability of the optical flow value at that location. A tight symmetrical likelihood translates to a good estimator. We also suggest that likelihoods should be propagated spatially in two steps before temporal propagation. Firstly, the *aperture* problem is addressed and secondly the *extended blank wall* problem is solved. Three additional observations about the Weiss-Fleet generative model [6] are that: (1) it is not necessary to include noise in the model as noiseless images produce similar likelihoods; (2) it is not clear that the exponential is necessary as part of the likelihood (i.e., the finite sampling window can be treated as a prior which only looks at slow speeds); and (3) we argue that uncertainties need to be propagated through time and maximization/minimization or the expected values of the likelihood may not be appropriate for motion estimates.

4 Condensation Approach

The Condensation algorithm [13], also called particle filtering, is usually used for tracking objects where the posterior probability function is not unimodal or can be modelled by a predefined function such as a Gaussian. The Condensation approach is useful when there are multiple hypothesis and it is necessary to propagate them across time. A Monte Carlo technique of factored sampling is used to propagate a set of samples through state space efficiently. The posterior probability $P(X_t \mid I(x,y,t))$, can be computed by using:

$$P(X_t \mid I(x,y,t)) = \frac{P(I(x,y,t) \mid X_t) P(X_t \mid X_{t-1})}{P(I(x,y,t))} \quad (9)$$

$$= \alpha P(I(x,y,t) \mid X_t) P(X_t \mid X_{t-1}) \quad (10)$$

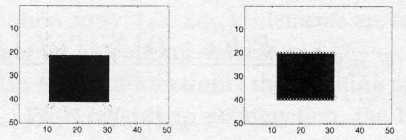
;where X_t expresses the state at time t. The prior $P(X_t \mid I(x,y,t-1))$ is inferred from predicting $P(X_{t-1} \mid I(x,y,t-1))$ through a temporal model $P(X_t \mid X_{t-1})$ which is used for computing the measurements (observations) $P(I(x,y,t) \mid X_t)$ (i.e., the likelihood), from which the posterior follows. The temporal model typically includes a deterministic drift component and a random diffusion component. It is also a set of samples $S_t = [s_1, s_2, ..., s_N]$ selected from S_{t-1} using a sample-and-replace scheme that are propagated. The posterior is only computed to an unknown scale factor α.

We adapt this terminology in the context of the SD (sum-difference) algorithm proposed by Camus [5]. The prior is expressed as the sampling window. The likelihood function is produced from the simulated possible SD motions. We only explore using a fixed lattice for computing the probability but this does not necessarily have to be the case and should not be. The lattice or random samples should be determined from the propagation based on the priors. The SD motions are for patches. The maximum displacement is used as a normalizing constant to produce probabilities. Also, the produced probabilities are further normalized so that the likelihood pdf (probability distribution function) sum is unity.

5 Sum-Difference Bayesian View

In order to adapt the Sum-Difference (SD) approach with a Bayesian perspective, it is necessary to investigate what form the likelihood functions in a condensation framework take. Firstly, With the SD approach, or for that matter, the SSD (Sum Difference Squared) approach, a decision is made on which is the best velocity vector to choose. This is what the Camus algorithm does in its basic form [5]. Embellishing the vectors by performing interpolation to get a more smooth vector field is also possible as an additional step. For a simple test scenario, we decided to look at the fields produced by a two-sequence image consisting of a somewhat noisy square (see Figure 3) moving in the direction of the top-right hand corner. Averaging was a 5x5 window and a possible 7x7 motions were simulated centred on the pixel of interest. Figure 4 shows the resultant flow field produced by a two image pair describing movement towards the up-right direction. Notice that some of the vectors appear to be quite arbitrarily chosen.

Alternatively, we can treat all the possible choices for



(a) (b)

Figure 3: **Moving Square data** : used for tests in Figures 4, 5, 6 and 7.

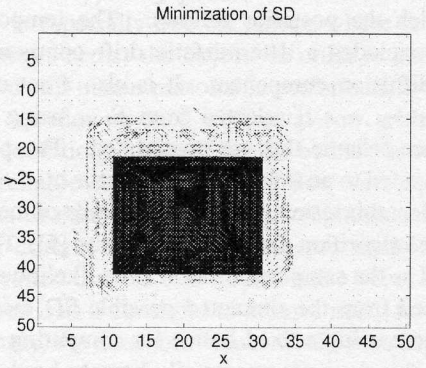


Figure 4: **Minimum Error:** using the SD approach for the temporal pair in Figure 3. This also can be expressed as finding the maximum probability.

each position as a pdf. The maximal possible change in intensity value is used as a scaling factor for describing these values as probabilities. Subsequently, the expected value of this pdf is computed for each position in the image. The resultant for the moving square is shown in Figure 5. The expected flow value appears to better meet our expectations. The flow at the corners tend to indicate what we expect from the top-right directed motion. In addition, motion along the horizontal and vertical edges also show the aperture problem, where the expected value is normal to the edge in question and the vertical component is undetermined.

Figure 6 illustrates what the SD difference values in probability form look like when computed at various locations in the image. Notice that the probabilities in 6a, 6b, and 6d show the aperture problem, while 6c shows the flow at the bottom right corner. The probability in 6c reflects our expectations but the flow in the expected value does not reflect this as shown in Figure 5. Figure 6e illustrates what the pdf looks like for the blank wall problem while 6f presents the pdf for what is expected as a zero flow. The pdfs for 6e and 6f are not distinguishable and it is only the context that differentiates them.

The likelihood function is traced from inside the square to outside the square going across the top-right hand corner

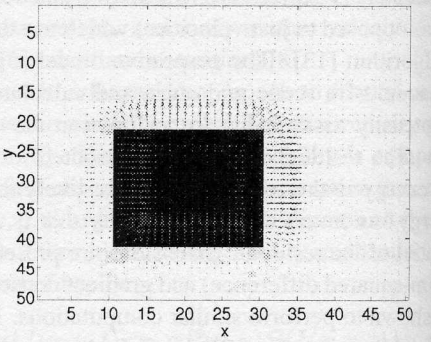


Figure 5: **Expected Optical Flow** computed at each sampled location for the temporal pair in Figure 3. The optical flow is computed by using the expected value of the local pdf.

in Figure 7. It is interesting to note how the function evolves along this gradient. The pdfs show high certainty that the flow in most of the sequence does not flow to the bottom left. The peak is visible in Figure 7b. It is highly distinctive and critical for helping decide the flow in the cases where the aperture problem exists.

The various models including the proposed generative models described earlier for modelling the likelihoods all assumed that the probability functions actually represented the residuals, or what can be called the noise. We added Gaussian noise to the entire image as shown in Figure 8. The max/min approach produces a completely different vector set in Figure 9 when compared to Figure 4. It is interesting to note that the expected flow field in the noisy image of Figure 10 is similar to the non-noisy image of Figure 5. Note also that both of these results are much better qualitatively than the Horn-Schunk algorithm [2] results in Figure 11.

We decided to look to at the pdfs computed on the noisy image at locations that were computed equivalently in the relatively non-noisy image in Figure 6a,b,c, and d. These are shown with the matching sub-indexes in Figure 12. It is interesting to note that the pdfs are almost identical in shape. Rather than attributing the residual to noise, it appears that the statistics of movement produce the residual.

In Figure 2, we used the SD min/max approach for computing the optical flow. Subsequently we thresholded the flow and diffused this flow by using Laplacian equation where the flow vectors that survived the thresholding step were used as boundary conditions. We were not interested in the actual flow vector values but rather the segmentation of moving from static objects. In the moving square scenario, flow is also detected on the moving background, i.e., the uncovering/covering of the background. If the background is not uniform as in the natural scene, these flow vectors are not present. This is also the case in Figure 2 where the flow

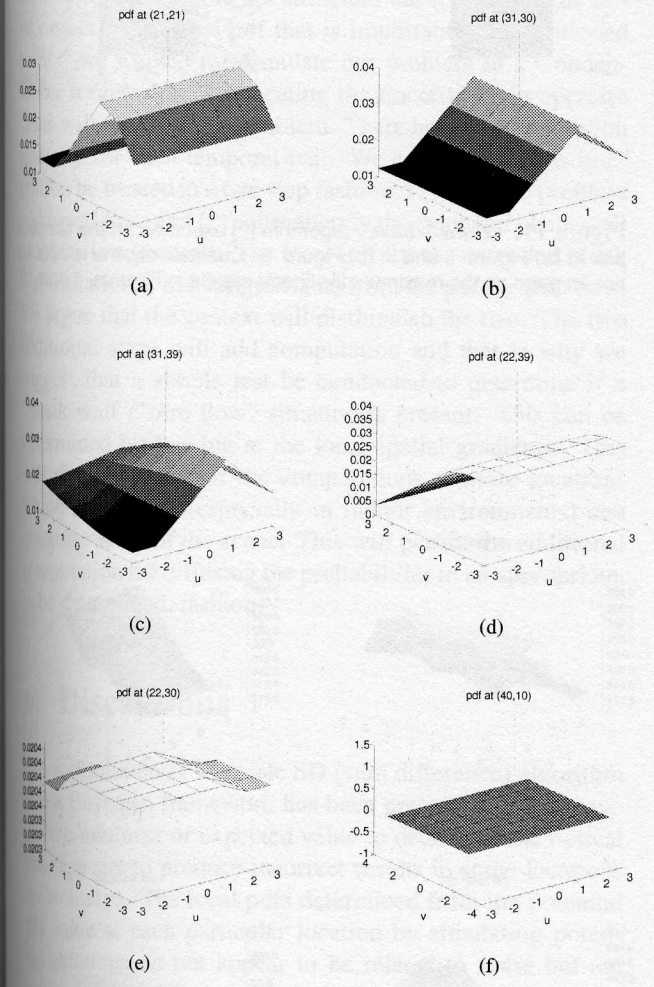


Figure 6: **Local Probabilities**: at various locations based on the stimulus presented in Figures 5 and 4. (a) is taken at location $(21,21)$; (b) is taken at location $(31,30)$; (c) is taken at location $(31,39)$; and (d) is taken at location $(22,39)$. Notice the aperture problem in (a), (b) and (d). (e) is taken at (22,30) and (f) is taken at (40,10). (e) shows the blank wall problem, while (f) shows the case of zero flow. Note that (e) and (f) are not distinguishable.

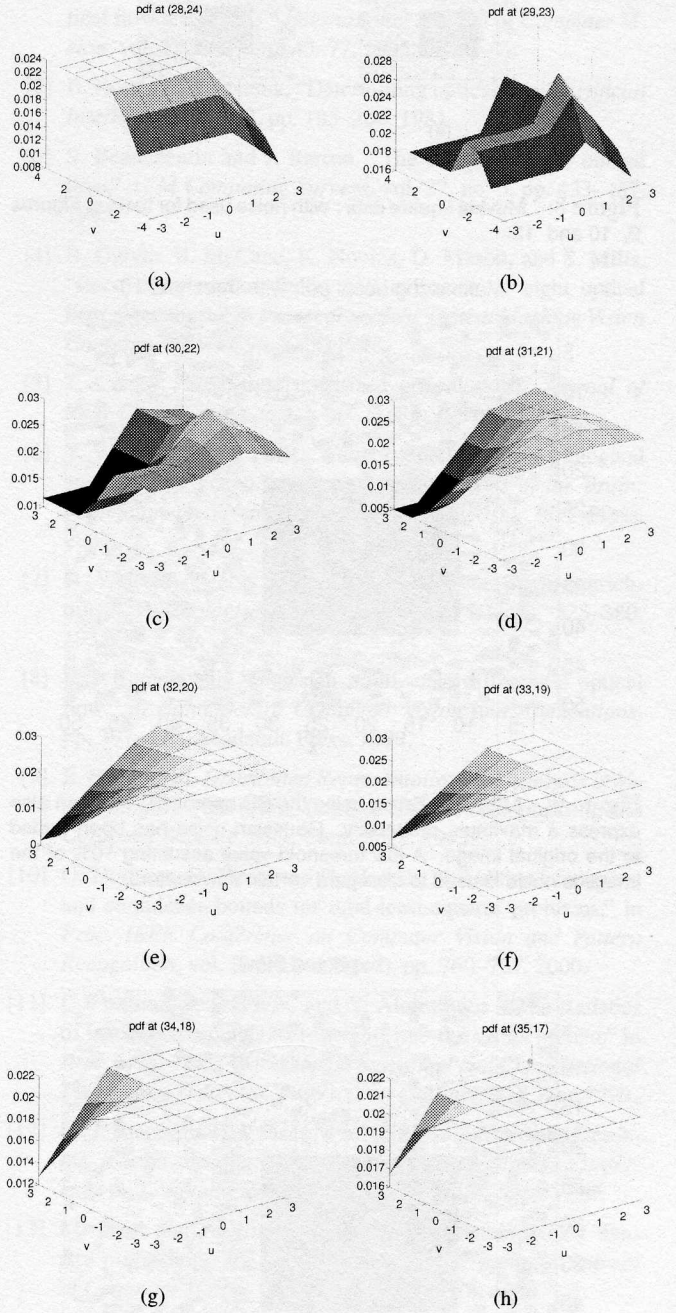
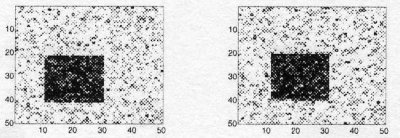


Figure 7: **Local Probabilities**: at various locations based on the stimulus presented in Figure 5. (1) is taken at location $(28,24)$; (2) is taken at location $(29,23)$; (3) is taken at location $(30,22)$; (4) is taken at location $(31,21)$; (5) is taken at location $(32,20)$; (6) is taken at location $(33,19)$; (7) is taken at $(34,18)$; and (8) is taken at $(35,17)$. Notice how the likelihood functions vary as one crosses the top-right corner of the square.



(a) (b)

Figure 8: **Moving Square data** : with noise used for tests in Figures 9, 10 and 12.

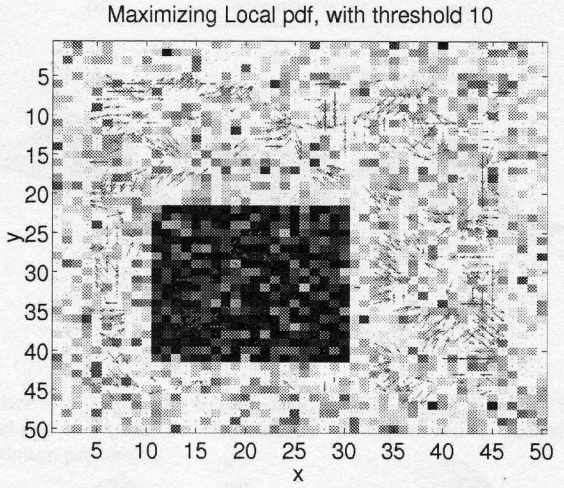


Figure 9: **Minimum Error.** using the SD approach. This can also express a maximum probability. Gaussian noise has been added to the original image. A low threshold value assuming 10% of the image is noise is used to disregard certain flow vectors.

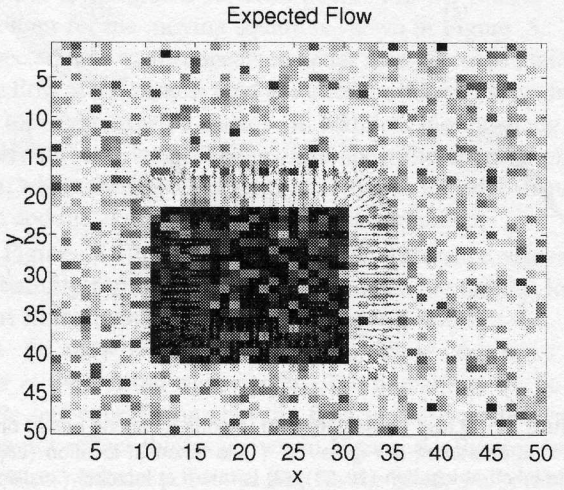
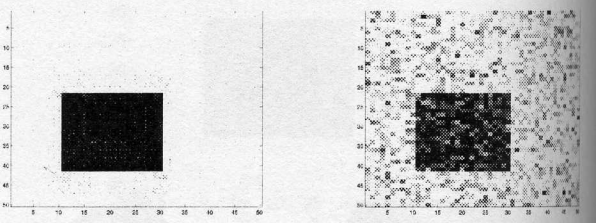
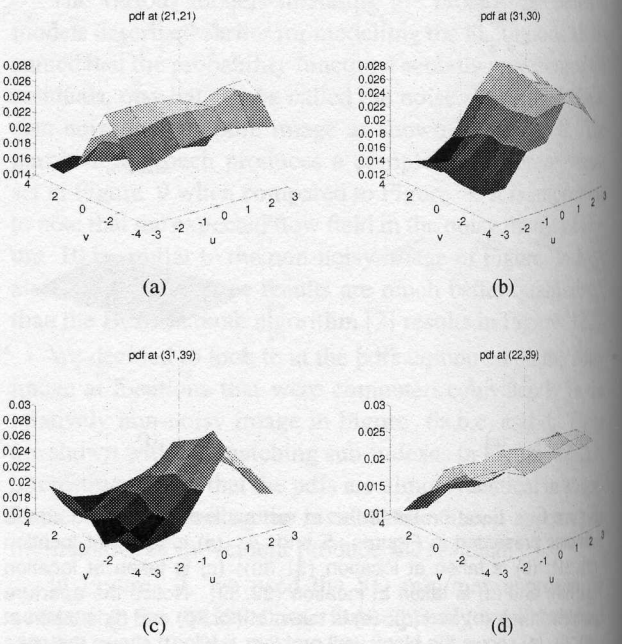


Figure 10: **Expected Optical Flow** computed at each sampled location of a noisy image. The optical flow is computed by using the expected value of the local pdf. The stimulus is a well defined square moving up towards the left.



(a) (b)

Figure 11: **Horn-Schunk** : algorithm [2] executed on the temporal pair in (a) Figure 3 and in (b) Figure 8. Qualitatively, the results are not as good as the maximum likelihood results in Figures 5 and 10.



(a) (b)

(c) (d)

Figure 12: **Local Probabilities for Noisy Image**: at various locations based on the stimulus presented in Figure 5. (1) is taken at location $(21, 21)$; (2) is taken at location $(31, 30)$; (3) is taken at location $(31, 39)$; and (4) is taken at location $(22, 39)$. Notice the aperture problem in (1), (2) and (4).

field extends beyond the border of the arm. The dynamics of the flow field at spatial discontinuities [12] is an area of further investigation. The evolution of the flow field pdf as shown in Figure 7 may give us some clues on how to proceed but we need to investigate more and various stimuli.

In addition, from the discussions above, we conclude that performing a straight min/max or expected value from the possible choices is not sufficient but it is rather the observation of the local pdf that is important. As mentioned earlier, we wanted to formulate our problem in a Condensation framework. Propagating the uncertainty in aperture cases will not solve the problem. There has to be integration spatially for each temporal pair. We suggest that this integration be treated in a two step fashion: (1) aperture problem addressed first; and (2) subsequently the extended blank wall problem is addressed. The local pdf for the extended blank wall situation is indistinguishable from the pdf for zero flow. We argue that the context will distinguish the two. The two additional steps will add computation and that is why we suggest that a simple test be conducted to determine if a "blank wall"/"zero flow" situation is present. This can be determined by looking at the local spatial gradients. This will alleviate some of the computations at these locations which sometimes (especially in indoor environments) can comprise most of the scene. This will permit the additional computation for diffusing the probabilities in an appropriate, to be determined, fashion.

6 Discussions

The formulation of a simple SD (sum difference) algorithm into a Bayesian framework has been presented. Simply using the min/max or expected value to determine the optical flow appears to produce incorrect results in some locations in the image. The local pdfs determined from the potential SD value at each particular location by simulating potential solutions do not appear to be related to noise but are more closely related to the underlying motions. We have proposed that the optical flow problem be cast in a Condensation framework. The aperture and extended blank wall problem will have to be better addressed with each temporal pair of images before temporal propagation takes place. A simple diffusion of the optical flow for motion segmentation was illustrated, but distinguishing flow of the foreground object and background at spatial discontinuities has yet to be properly addressed.

Acknowledgments

The author expresses thanks to funding from the National Science and Engineering Research Council (NSERC).

References

[1] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1995.

[2] B. Horn and B. Schunk, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.

[3] S. Beauchemin and J. Barron, "The computation of optical flow," *ACM Computing Surveys*, vol. 27, no. 3, pp. 433–467, 1995.

[4] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, "Recovering motion fields: An evluation of eight optical flow algorithms," in *Proceedigs of the British Machine Vision Conference BMVC98*, Sept. 1998.

[5] T. Camus, "Real-time quantized optical flow," *Journal of Real-Time Imaging*, vol. 3, pp. 71–86, 1997.

[6] Y. Weiss and D. J. Fleet, "Velocity likelihoods in biological and machine vision," in *Probabilistic Models of the Brain: Perception and Neural Function*, pp. 81–100, MIT Press, 2001.

[7] H. Wallach, "Ueber visell whargenommene bewegungrichtung," *Psychologische Forschung*, vol. 20, pp. 325–380, 1935.

[8] E. P. Simoncelli, "Bayesian multi-scale differential optical flow," in *Handbook of COmputer Vision and Applications*, pp. 397–422, Academic Press, 1999.

[9] E. Simoncelli, *Distributed Representation and Analysis of Visual Motion*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, 1993.

[10] O. Nestares, D. Fleet, and D. Heeger, "Likelihood functions and confidence bounds for total-least-squares problems," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, (Hilton Head), pp. 760–767, 2000.

[11] C. Fernmuller, R. Pless, and Y. Aloimoinos, "The statistics of visual correspondence: Insights into the visual system," in *Proc. of the IEEE Workshop on Statistical and Computational Theories of Vision (SCTV99)*, (Fort Collins, CO), June 1999.

[12] M. J. Black and D. J. Fleet, "Probabilistic detection and tracking motion discontinuities," in *1999 International Conference on Computer Vision (ICCV99)*, (Corfu), Sept. 1998.

[13] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.